PKZIP® 6.0
Command Line
Getting Started Manual

Table of Contents

Operating System Requirements

The minimum system requirements for PKZIP Command Line for Unix are as follows:

Platform:	Minimum System Requirements:
Sun Solaris® 2.6 and later	32 MB RAM; 1.5 MB hard disk space for programs and minimal workspace. UltraSPARC processor.*
IBM AIX® 4.3 and later	32 MB RAM; 1.5 MB hard disk space for programs and minimal workspace.
HP-UX® 10.20 and later	32 MB RAM; 1.5 MB hard disk space for programs and minimal workspace.
Linux® running the 2.4 or later kernel over an Intelx86 processor	32 MB RAM; 1.5 MB hard disk space for programs and minimal workspace. glibc-2.24

* To run the program pkzipc, you must have an UltraSPARC processor. If you have an older SPARC processor, you should run pkzipc.v7 instead.

Release Notes

PKZIP 6.0 Command Line provides a number of new features. Refer to the file WHATSNEW.HTML for a summary of these features. Several important changes related to configuring the program are noted below.

- The program name used in prior releases has been changed. Earlier versions of PKZIP Command Line used the program name **PKZIP25**. This has been changed to **PKZIPC**.

- Two new DOS translation programs are provided in this release to support using DOS 2.04g command line syntax to run the new PKZIP Command Line program. Refer to the section *Configuring Your System to Use the DOS Translation Set* in this document for more information on this feature.

Installing and Configuring PKZIP

Installing from a PKSFX file:

If you are installing a version of the software downloaded from the Internet you will receive a PKSFX self-extracting file. The Installer is packaged inside the PKSFX file. Run the PKSFX file to start the installation. It will extract the files needed to install the software and it automatically starts the setup program. Follow the instructions provided by the setup program to complete the installation.

We encourage you to make a backup of your PKSFX file, or of the contents of the intermediate folder, to ensure you will be able to reinstall this product if you ever need to in the future.

Setting PKZIP in the Path

To access PKZIP from any directory without typing a path, specify the PKZIP install (destination) directory in your system's path environment variable. The path specification is typically located in a start-up file such as .profile or .login.

To add the PKZIP installation directory to your path:

1. Open your start-up file using a text editor such as vi, Pico, and Emacs.

2. What you do next depends on the shell you are using:

   If you are using the Korn Shell (**ksh**) or the Bourne Shell (**sh**), look for a line similar to the one below in your .profile file.

   > **PATH=/usr/bin:**

   If you are using the C Shell (**csh**), look for a line similar to the one below in your .login file:

   > **setenv PATH /usr/bin:**

   or:

> **set path=(/usr/bin .)**

A colon separates path designations in the PATH= and setenv PATH examples while a single space separates path designations in the set path= example.

3. Add the PKZIP installation directory (using a colon or space to separate as appropriate) to the PATH=, setenv PATH, or set path = lines. If, for example, you installed PKZIP in the /usr/local/pkware/pkzipc directory, add the following:

> **/usr/local/pkware/pkzipc**

The path you specify depends on the location of the pkzipc file. If you use the C Shell, proceed to step 5. Korn and Bourne Shell users go to step 4.

4. Verify that the line below exists in your start-up file:

> **export PATH**

If it does not exist, add it after the PATH= line.

5. Save and exit the file.

6. Reset your current environment settings. This can be accomplished by logging off your account. The next time you log on to your account, you can run pkzipc without specifying a path, regardless of the current working directory.

Creating the Tutorial Directory and Files

To follow the tutorials in the **PKZIP User's Manual**, create a "working" directory. This is a "temporary" directory used only for the tutorials. Creating this temporary directory, helps ensure that your permanent directories and files are not deleted or damaged while you practice with PKZIP.

To create your workspace:

1. Change to the directory where you installed PKZIP, such as /usr/local/bin/pkware/pkzipc.

2. Run the tutorial.sh file. Type the following and press **Enter**:

   sh tutorial.sh

   Running the tutorial.sh file creates a special tutorial directory as a subdirectory of the install directory such as /usr/local/bin/pkware/pkzipc/tut. Additionally, it copies several *test* files for you to use with the tutorials.

3. Go to the directory that was created in the previous step. Type the following and press **Enter**:

   cd tut

4. To confirm that the files were copied, type the following and press **Enter**:

   ls -l

   A file list similar to the following appears:

   ```
   -rw-r--r--  1 user    pkware     43326 Jul  1 02:51 black.tut
   -rw-r--r--  1 user    pkware      4445 Jul  1 02:51 blue.fil
   -rw-r--r--  1 user    pkware      5777 Jul  1 02:51 brown.doc
   -rw-r--r--  1 user    pkware     43326 Jul  1 02:51 gold.tut
   -rw-r--r--  1 user    pkware       582 Jul  1 02:51 green.doc
   -rw-r--r--  1 user    pkware       582 Jul  1 02:51 orange.fil
   -rw-r--r--  1 user    pkware     43326 Jul  1 02:51 pink.tut
   -rw-r--r--  1 user    pkware       582 Jul  1 02:51 purple.txt
   -rw-r--r--  1 user    pkware      4445 Jul  1 02:51 red.txt
   -rw-r--r--  1 user    pkware      4445 Jul  1 02:51 tan.txt
   -rw-r--r--  1 user    pkware      5777 Jul  1 02:51 white.doc
   -rw-r--r--  1 user    pkware      5777 Jul  1 02:51 yellow.doc
   ```

   You are now ready to use these files with the PKZIP practice tutorials. Refer to *Chapter 2 - The Basics* in the **PKZIP User's Manual** for more information.

Configuring DOS Translation Programs

The DOS translation programs provided with this release allow you to use the same program names and command line switches as in the original version of PKZIP for DOS. If you are familiar with the command line switches from PKZIP for DOS, you may want to use these translation programs to make using PKZIP Command Line easier. They provide a quick way to start using the new program without conflicting with your current methods or having to immediately learn the new command switches.

The DOS translation programs are installed to the installation directory.

After you have run the installation, your system will be configured to use the translation versions of **PKZIP** and **PKUNZIP** to run PKZIP Command Line. Refer to Appendix E in the PKZIP Command Line User's Manual for more information on the commands supported by the DOS translation programs.

The PKZIPCFG Environment Variable and pkzip.cfg File

To configure and customize commands and options in PKZIP, you can use the *configuration* command on the PKZIP command line, or directly edit the configuration file (pkzip.cfg) with a text editor. The recommended method for configuring and customizing PKZIP commands and options is via the command line using the *configuration* command.

To set the PKZIPCFG environment variable:

1. Open your start-up file using a text editor such as vi, Pico, and Emacs.

2. What you do next depends on the shell you are using:

If you use the Korn Shell (**ksh**) or the Bourne Shell (**sh**), add the following lines to your .profile file:

> **PKZIPCFG=<path to the pkzip.cfg file>**
>
> **export PKZIPCFG**

If you use the C Shell (**csh**), add the following line to your .login file:

> **setenv PKZIPCFG <path to the pkzip.cfg file>**

The path you specify for pkzip.cfg depends on the location of the file. The pkzip.cfg file typically resides in the PKWARE installation directory such as /usr/local/bin/pkware/pkzipc/pkzip.cfg.

3. Save and exit the file.

4. Reset your current environment settings. This can be accomplished by logging off your account. The PKZIPCFG variable will be set the next time you log on to your account.

   You can now modify defaults in the PKZIP Configuration Settings file using the *configuration* command. For information on setting default values in PKZIP, refer to *Chapter 6 - Changing Defaults Using the Configuration File* in the **PKZIP User's Manual**.

Note: If you do not set a PKZIPCFG environment variable, PKZIP creates a default pkzip.cfg file in your home directory. This occurs when the configuration command is specified on the command line and PKZIP is unable to find the file.

Editing the pkzip.cfg File

You have the option of modifying the pkzip.cfg file directly with a text editor. You can modify date formats, as well as command line argument options. The default pkzip.cfg file used by PKZIP looks similar to the following:

```
# PKZip configuration file format section
:FORMATS
iDate=0
sDate=/
# PKZip configuration file option section
:OPTIONS
```

Entries in the pkzip.cfg that follow a "#" character are treated as comments. Entries that follow the ":FORMATS" designation modifies date format settings. Entries that follow the ":OPTIONS" designation modify command argument options.

Configuring Date Order in the pkzip.cfg File

PKZIP uses MMDDYY as the default date order. You can specify the date order used by PKZIP if you modify the *idate* option after ":FORMATS".

The available *idate* sub-options are listed in the following table:

Sub-Option	To:	For example:
0	Set PKZIP date format to **MMDDYY**.	idate=0
1	Set PKZIP date format to **DDMMYY**.	idate=1
2	Set PKZIP date format to **YYMMDD**.	idate=2

Specifying idate=1 in your pkzip.cfg file, for example, sets the default PKZIP date format to DDMMYY. File information that appears in the *-view* option is presented in this format. Any files that you subsequently add or extract, using such options as *-after* and *-before,* require the use of this particular format in order for PKZIP interpret it correctly.

Note: The *locale* option on your command line overrides an *idate* specification in your pkzip.cfg. Additionally, the date format listed in the PKZIP Configuration Settings screen does not necessarily reflect your *idate* setting. The date format listed in PKZIP Configuration Settings screen lists your *locale* setting, which is independent of the *idate* setting. For more information on the *locale* option, refer to the **PKZIP User's Manual**.

PKZIP 6.0 Command Line – Getting Started Manual

Configuring the Separator Character in the pkzip.cfg File

You can specify the field separator PKZIP uses by modifying the **sdate** option in the pkzip.cfg. By default, PKZIP uses a "/" character to separate the date fields. If, for example, you want to change the field separator character from a "/" to a "-", you could add the following line to the pkzip.cfg after the ":FORMATS" designation:

sdate=-

Specifying **sdate=-** in the pkzip.cfg file, for example, sets the default PKZIP date format to DD-MM-YY. File information that appears via the **-view** option is presented in this format.

Note: The **idate** and **sdate** options are only configurable by editing the pkzip.cfg file with a text editor.

Configuring Command Line Arguments Options in the pkzip.cfg File

You may specify default command line argument options in PKZIP by adding the appropriate command/option and sub-option (if applicable) after the ":OPTIONS" designation in the pkzip.cfg. If, for example, you wish to set the default for all **add** operations to **update**, add the following line to the pkzip.cfg file under the ":OPTIONS" designation:

-add=update

For more information on the available commands/options and sub-options, see the **PKZIP User's Manual**.

Sample pkzip.cfg File

Assume that you wish to instruct PKZIP to use DDMMYY as the default date order, as well as use the "-" character as the separator character. Additionally, assume that you wish to the set the default command line argument for the **filetype** option to process all file types. To do this, edit the pkzip.cfg file with a text editor. Specifically, modify the

idate and *sdate* options under the ":FORMATS" designation and add the appropriate command line argument, such as **-filetype=all,** under the ":OPTIONS" section. Your pkzip.cfg file should look similar to the following example:

```
# PKZip configuration file format section
:FORMATS
iDate=1
sDate=-
# PKZip configuration file option section
:OPTIONS
-filetype=all
```

Setting the PKSFXDATA Environment Variable

PKZIP includes an external data file called pksfxs.dat. The pksfxs.dat file allows you to create self-extractors. The *listsfxtypes* option, described in Appendix A in the **PKZIP User's Manual**, displays a list of the types of SFX files that can be created with PKZIP. This file is usually kept in the source directory with the pkzipc binary.

You can change the location of the pksfxs.dat file on your system with the help of an environment variable called PKSFXDATA. To set the PKSFXDATA environment variable, do the following:

1. Using a text editor such as vi, Pico, Emacs, open your start-up file.

2. What you do next depends on the shell you are using:

If you are using the Korn Shell (**ksh**) or the Bourne Shell (**sh**), add the following lines to your .profile file:

> **PKSFXDATA=<path to the pksfxs.dat file>**

> **export PKSFXDATA**

If you are using the C Shell (**csh**), add the following line to your .login file:

setenv PKSFXDATA <path to the pksfxs.dat file>

The path you specify depends on the location of the pksfxs.dat file. The pksfxs.dat file typically resides in the PKWARE installation directory (/usr/local/bin/pkware/pkzipc).

3. Save and exit the file.

4. To reset your current environment settings, log off your account. The PKSFXDATA variable will be set the next time you log on to your account.

The PKSFXDATA environment variable allows you to place the pksfxs.dat file in any directory without affecting SFX functionality in PKZIP. For example, if you placed the pksfxs.dat file on a different file system, you could then use the PKSFXDATA environment variable to point to that file. Assuming that your computer has access to the specified file system, there would be no need to copy the pksfxs.dat file to your current file system. PKZIP attempts to locate the pksfxs.dat file by looking in the environment variable path first, current path second, and command line path last.

Using Wildcards with PKZIP on Unix

There are two factors that control how wildcards are processed by PKZIP running on a Unix system: your shell configuration and the format of the wildcards on your PKZIP command line.

In instances where your shell is set up to automatically expand wildcards, PKZIP is given a list of explicit filenames (generated by the shell) that match the specified wildcard pattern. Because of the way PKZIP interprets this explicit file list, it may not archive all files when the *recurse* or *directories* option is specified on the command line. As such, Unix users should use the *include* option, or place quotation marks around wildcard designations. This

bypasses automatic wildcard expansion by your shell which may, in turn, restrict your pattern search.

Digital Certificates

PKZIP allows you to digitally sign the individual files archived in a .ZIP file (as well as the Central Directory) and subsequently authenticate those files upon extraction. PKZIP signing functionality is based on the X.509 certificate standard and is therefore compatible with authenticity functionality in other applications such as Microsoft's Internet Explorer. Signing a .ZIP file allows you to detect whether a .ZIP file's integrity has been compromised. Before configuring PKZIP to sign files, you must first have a digital certificate to use for signing. These certificates must be a minimum of 1024-bit RSA format. Digital certificates are available from a variety of certificate authorities. Visit our web site for information on obtaining a certificate:

http://www.pkware.com/catalog/certificate.htm

Installing the ROOT, CA, and SPC Certificates:

The first thing you must do, if you want to verify signatures or sign .ZIP files is to install the ROOT and CA certificates. You might also want to include the SPC certificates, if any. To accomplish this, we suggest using the authenticode tools from Microsoft on a Windows computer with Internet Explorer 4 or better installed. If you do not already have these tools, Microsoft has them on their website.

If you have Internet Explorer 4 installed, visit the following URL:

http://msdn.microsoft.com/MSDN-FILES/027/000/218/codesign.exe

If you have Internet Explorer 5 installed, visit this URL instead:

http://msdn.microsoft.com/MSDN-FILES/027/000/219/codesign.exe

If you have IE 6, you will need to download Microsoft's ActiveX SDK to obtain the authenticode tools.

Unfortunately, no one has provided tools to obtain these certificates from other systems, such as Netscape running on your UNIX system.

1. Run the following commands (on the windows computer):

   certmgr –add -7 –all –s Root Root.p7

   certmgr –add -7 –all –s CA CA.p7

   certmgr –add -7 –all –s SPC SPC.p7

2. Transfer those three files to your home directory on the target UNIX system.

3. Become the superuser (root, use the su command).

4. Create the following directories:
   /usr/local/certificates
   /usr/local/certificates/CA
   /usr/local/certificates/ROOT
   /usr/local/certificates/SPC

5. Copy the p7 files to the appropriate directories:

   cp CA.p7 /usr/local/certificates/CA

   cp Root.p7 /usr/local/certificates/ROOT

   cp SPC.p7 /usr/local/certificates/SPC

6. Make sure the files and directories have read-only access.

   chmod -R 0555 /usr/local/certificates

7. Exit the superuser shell.

   exit

   Note: If you are not an administrator, ask your system administrator to do this for you. If the system administrator is unwilling, you can instead create the directories in your home directory. For example, if your home directory is /home/todd, you would create the following directories:

   /home/todd/certificates
   /home/todd/certificates/CA
   /home/todd/certificates/ROOT
   /home/todd/certificates/SPC

   After you have created those directories, you need to set

some environment variables. Users of sh based shells (sh, ksh, bash, zsh, etc.) would run the following commands:

ROOT_CERTIFICATES=/home/todd/certificates/ROOT

export ROOT_CERTIFICATES

CA_CERTIFICATES=/home/todd/certificates/CA

export CA_CERTIFICATES

SPC_CERTIFICATES=/home/todd/certificates/SPC

export SPC_CERTIFICATES

Users of csh or tcsh would run:

setenv ROOT_CERTIFICATES /home/todd/certificates/ROOT

setenv CA_CERTIFICATES /home/todd/certificates/CA

setenv SPC_CERTIFICATES /home/todd/certificates/SPC

You can put those commands in your login file (.login for sh users, .cshrc for csh users) to always have them available when you log in.

Using Wildcards with PKZIP on Unix

By placing a wildcard pattern in quotation marks, ("*") you are instructing the shell to pass the object as an argument into PKZIP. Consequently, PKZIP processes the wildcard expansion as opposed to the shell. This is equivalent to using the *include* option on your command line; for example, include="*.txt". If your Unix shell is not configured to expand wildcards, such modifications are not necessary.

PKZIP can interpret and expand the following wildcard patterns:

1.   *

2.   *<pattern>   (*.txt, *f.txt)

3.   <pattern>*   (h*, file.f*)

4. <pattern>*<pattern> (a*.txt)

5. *<pattern>* (*.*, *ab*)

Authenticity Verification

PKZIP allows you to embed an electronic signature with files stored in a .ZIP archive, and subsequently, authenticate those files on extraction. PKWARE Authenticity Verification (AV) information allows you to detect whether a .ZIP file's integrity has been compromised. You may use the **authenticity** command line option to instruct PKZIP to embed AV information with your .ZIP files. However, before you can use this option, you must first configure PKZIP for AV. To configure PKZIP for AV, complete the following steps:

▪ Complete the Application for Authenticity Verification (authveri.txt) found in the PKZIP 6.0 Command Line installation directory (/usr/local/bin/pkware/pkzipc) or on page 20 of this manual. You may also access the Application for Authenticity Verification via the web at the following URL:

http://www.pkware.com/authenticity

▪ Complete your application at our web address or fax (+1-414-354-8559) your completed application to PKWARE. PKWARE will process your application and send you a confirmation message via the United States Postal Service or FAX containing two license keys. These license keys are required to configure PKZIP for Authenticity Verification.

▪ After you receive your confirmation message, go to the command prompt and change to the PKZIP installation directory. Run the putav program by typing the following:

putav

After you press **Enter**, you are prompted to enter your company name and the two license keys as they appear in the confirmation message you received from PKWARE. Screen output similar to the following appears:

```
This program will configure PKZIP for Authenticity
Verification. To proceed, complete and submit the Application
for Authenticity Verification located in the PKZIP installation
directory (authveri.txt) or in the PKZIP Getting Started
Manual. You may also access the Application for Authenticity
Verification via the World Wide Web at the following URL:
http://www.pkware.com/authenticity
---------------------------------------------------------------
PKWARE will process your application and send you a
confirmation message via the United States Postal Service
or FAX. This confirmation message will contain two serial
numbers required to configure PKZIP for Authenticity
Verification.
---------------------------------------------------------------
Press Ctrl-C to abort or return to continue
```

Press **Enter**. The following appears:

```
PUTAV Ver 4.0 - Put Authenticity Verification in PKZIP
Copyright 1990-2000 PKWARE, Inc. All rights reserved.

Enter company name exactly as it appears on the PKWARE documentation.
Company Name: Vandelay Industries
Enter license keys exactly as they appear on the PKWARE documentation.
License key 1: 8SU2l4X
License key 2: A7YG15W

Working, done.

License key installation complete. Check value: 31f50e0d
```

You must enter your company name and the two license keys exactly as they appear in the confirmation message. The check value displayed on your screen should match the check value specified in your confirmation message. If they do not match, repeat step 3. For more information on how PKZIP looks when you extract files that have been digitally signed with an AV signature, refer to page 95 in the User's Manual. For AV related warning messages, refer to Appendix B in the User's Manual.

Notes

You may include additional information such as telephone number and address with your AV string. To do this, place an ASCII text file containing this additional information named **AVEXTRA.TXT** in your current working directory.

Setting the PKAVEXTRA Environment Variable

To include additional information with your AV string, you can place the text in a file and specify the file location in the PKAVEXTRA Environment variable.

You can specify the location of the pkavextra.txt file on your system with the help of an environment variable called PKAVEXTRA. To set the PKAVEXTRA environment variable, do the following:

1. Using a text editor such as vi, Pico, Emacs, open your start-up file .profile or .login.

2. What you do next depends on the shell you are using:

    If you are using the Korn Shell (**ksh**) or the Bourne Shell (**sh**), add the following lines to your .profile file:

    > **PKAVEXTRA=<path to the pkavextra.txt file>**
    >
    > **export PKAVESTRA**

    If you are using the C Shell (**csh**), add the following line to your .login file:

    > **setenv PKAVEXTRA <path to the pkavextra.txt file>**

    The path you specify depends on the location of the pksfxs.txt file.

4. Save and exit the file.

5. To reset your current environment settings, log off your account. The PKAVEXTRA variable will be set the next time you log on to your account.

Application for Authenticity Verification

In order to receive (free of charge) license keys to use with the PUTAV program, please fill out and return this form to PKWARE. When the enclosed information is verified, you will receive your license keys from PKWARE. Please note that this does not constitute a license from PKWARE to distribute the executable program PKZIP or self-extracting PKSFX files for commercial purposes. See LICENSE.TXT or contact PKWARE, Inc.,for more information about distribution licenses for the software.

Please read the entire section on Authenticity Verification before proceeding.

1) Please specify your registration information:

Name:_____

E-mail Address:_____

Company:_____

Title:_____

Address:_____

City:_____ **State/Province:**_____ **ZIP/Postal Code:**_____

Country:_____

Telephone #:(_____)_____

Fax #:(_____)_____

Product Name:_____ **Version:**_____

Approximate date of registration/purchase:_____

2) Please specify the name that you want displayed by PKZIP when an authentic file created by you is verified (51 characters max; Case Sensitive).

⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜

⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜

⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜⬜
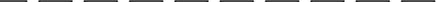
*** NOTE: PKZIP displays AV text on ONE line.

Since the purpose of the Authenticity Verification feature is to help provide detection of tampered, hacked or virus infected files, we ask that you provide only your company's name or individual's name for the AV string.

The "AVEXTRA.TXT" feature was designed to additionally display telephone numbers, addresses, email addresses, and similar information.

Strong Encryption

PKWARE has added the ability to use strong encryption when compressing files. This new feature adds a higher level of security for your data. If your version of PKZIP is licensed to use this feature, you will be able to choose from several strong encryption algorithms to encrypt your files. Strong encryption is available as part of the optional security module. If your copy of PKZIP is not licensed to use this feature, contact PKWARE at 1-414-354-8699, or visit www.pkware.com, for information on how to activate strong encryption.

With strong encryption you can encrypt your files using AES 128, 192, 256 bit algorithms. You can configure PKZIP to use any of these algorithms.

    AES – 128 bit
    AES – 192 bit
    AES – 256 bit
    3DES – 168 bit
    3DES – 112 bit
    DES
    RC4 – 128 bit
    RC4 – 64 bit
    RC4 – 40 bit
    RC2 – 128 bit
    RC2 – 64 bit
    RC2 – 40 bit

To use strong encryption, specify cryptalgorithm or recipient options when you create your .ZIP files. Refer to online help.

> **IMPORTANT:** PKWARE has introduced support for the use of strong encryption with .ZIP files. If you plan to share the .ZIP files you create using this strong encryption feature with others, you should first make sure the recipients have a compatible version of PKZIP 6.0 so they can decrypt your files.

The strong encryption feature lets you choose to use either a traditional password or an X.509 digital certificate for encryption. If you plan to use a digital certificate, you must first obtain and install a valid digital certificate onto your system. Visit PKWARE's website for information on where you can get a Digital Certificate.

File Capacities

The original .ZIP file format has faithfully met the needs of computer users since it was introduced by PKWARE in 1989. As computer technology has advanced over time, storage capacities have increased dramatically. These increases make the numbers and sizes of files that seemed unimaginable 10 years ago a reality today. To extend the utility of the .ZIP file format to meet these changing system needs, PKWARE has extended the .ZIP file format to support more than 65,535 files per archive and archive sizes greater than 4 Gigabytes (GB).

The specification for the .ZIP file format has been publicly available and distributed by PKWARE in a file called APPNOTE.TXT. This file documents the internal data structures and layout that define a .ZIP archive. The extensions introduced by PKWARE fully support all the features of your existing archives and newer versions of PKZIP that support these new extensions will continue to read all of your current archives.

Prior to the 4.5 release of PKZIP Suite, versions of PKZIP were limited to storing no more than 65,535 files in a .ZIP archive. Earlier versions of the original PKZIP for DOS could (and still can) store no more than 16,383 files per .ZIP archive.

Another limitation that existed prior to the 4.5 version of PKZIP Suite was that a single .ZIP archive could not be larger than 4 GB (4,294,967,295 bytes). Earlier versions of PKZIP for DOS cannot process .ZIP archives larger than 2 GB (2,147,483,647 bytes).

The extended .ZIP file format specification available with PKZIP Suite 4.5 and later supports creating .ZIP archives containing over 4 billion files and having sizes larger than 9 quintillion bytes. These are only theoretical limits and most computer systems in common use today do not have enough storage capacity or available memory to create and store .ZIP archives approaching these limits.

The practical limits imposed by a typical computer in use today and configured with 128MB to 256MB of memory will support compressing up to approximately 262,144 files. Compressing this number of files can take a long time. Since it is not practical to reach the theoretical limits supported by the extended .ZIP file format, PKZIP does not currently compress or extract more than 2,147,483,647 files.

The actual limits on the numbers of files and sizes of archives can vary depending on the operating system you are using. The tables below list the capacities available in PKZIP for different operating systems.

Windows NT/2000/XP, Unix, Linux	Current Version
Archive size	9 EB *
Number of files in archive	2 G **
Number of segments for spanned / split archives	4 G
Size of segments for spanned and split archives	4 GB
PKSFX size	2 GB
PKSFX Patch size	2 GB

Windows 95/98/Me	Current Version
Archive size	4 GB *
Number of files in archive	2 G **
Number of segments for spanned / split archives	4 G
Size of segments for spanned and split archives	4 GB
PKSFX size	2 GB
PKSFX Patch size	2 GB

* Actual archive size depends on available storage space - the theoretical limit is 9 EB.

** Actual number of files may vary depending on available memory and resources. The theoretical limit is actually 4 G but the practical limit for most current system configurations is about 262,144 files.

Your available system resources and system settings limit the performance you can expect from PKZIP when processing large numbers of files or large archives. If you are compressing large numbers of files on a computer with insufficient memory, you can expect slow, or possibly incomplete processing if your available memory is depleted.

When compressing large files, it is a good idea to make sure you have your temporary folder set to a location on a drive with sufficient disk space available to process large files. It is currently not recommended to use Drag-and-Drop with very large files or with very large numbers of files due to the additional system overhead required when Windows processes the files.

PKSFX self-extracting file sizes are limited to the size of an executable program file supported by the underlying operating system. This currently is 2 GB for most operating systems. This means you should not create a PKSFX file that is larger than 2 GB. PKZIP will be able to create PKSFX files larger than this limit, however, they won't run on most operating systems. You can create and run split PKSFX files that are larger than 2 GB as long as each split segment is not larger than 2 GB.

Support for more than 65,535 files in an archive and archive sizes greater than 4 GB are features not available in versions of PKZIP released prior to PKZIP Suite 4.50. If you create archives that take advantage of these new features, you should be aware that older versions of PKZIP will not be able to recognize all of the files in the new archives you create that exceed the capacities of the older programs. If you plan to send a large archive to a friend or associate, they will need a compatible version of PKZIP in order to extract the contents of the file(s) you send.

Specific PKWARE products that do support these new features are:

PKZIP V6.0 for Unix Command Line
PKZIP V6.0 for the X Window System
PKZIP V6.0 for Windows
PKZIP V6.0 Command Line
PKZIP Explorer V6.0

Versions of PKZIP prior to those noted above will not recognize these new features and will be unable to view or extract any files in your archives that are dependent on these features. Also, any .ZIP compatible programs you may be using from other companies will not be able to access all of the contents of your large archives. They may report that an archive is too large, or they may incorrectly report that the archive has errors. To ensure access to data in your large archives, always use genuine PKZIP.

Note: 4 GB is equal to 4,294,967,295 bytes.
9 EB is equal to 9,223,372,036,854,775,807 bytes.

Encoded and Archive File Support

PKZIP allows you to extract files from encoded or Tar and GZIP archived data. PKZIP will decode and extract files created with the following encoding and archive schemes:

Encoded File Type	Common File Extensions
UUEncode	*.UUE
XXEncode	*.XXE
BinHex	*.HQX
MIME	*.MIM *.MME
Tar	*.TAR
GZIP	*.GZ *.TGZ
BZIP2	*.BZ *.bz2

In addition to decoding and processing the above file types, PKZIP allows you to encode .ZIP files in the UUEncode format. To do so, specify the *encode* option on your command line as in the following example:

pkzipc -add -encode test.zip *